Evaluation of VOCALISE under conditions reflecting those of a real forensic voice comparison case (forensic_eval$_0$1)

Kelly, Finnian ; Fröhlich, Andrea ; Dellwo, Volker ; Forth, Oscar ; Kent, Samuel ; Alexander, Anil

DOI: https://doi.org/10.1016/j.specom.2019.06.005

Evaluation of VOCALISE under conditions reflecting those of a real forensic voice comparison case (forensic_eval_01)

Finnian Kelly , Andrea Fröhlich , Volker Dellwo , Oscar Forth , Samuel Kent , Anil Alexander

PII: S0167-6393(19)30018-4
DOI: https://doi.org/10.1016/j.specom.2019.06.005
Reference: SPECOM 2652

To appear in: *Speech Communication*

Received date: 14 January 2019
Revised date: 22 May 2019
Accepted date: 26 June 2019

Please cite this article as: Finnian Kelly , Andrea Fröhlich , Volker Dellwo , Oscar Forth , Samuel Kent , Anil Alexander , Evaluation of VOCALISE under conditions reflecting those of a real forensic voice comparison case (forensic_eval_01), *Speech Communication* (2019), doi: https://doi.org/10.1016/j.specom.2019.06.005

Evaluation of VOCALISE under conditions reflecting those of a real forensic voice comparison case (*forensic_eval_01*)

Finnian Kelly[1,2], Andrea Fröhlich[3], Volker Dellwo[4], Oscar Forth[1], Samuel Kent[1], and Anil Alexander[1]

{finnian|oscar|sam|anil@oxfordwaveresearch.com}, andrea.froehlich@for-zh.ch, volker.dellwo@uzh.ch

[1] Oxford Wave Research Ltd, Oxford, United Kingdom.
[2] Center for Robust Speech Systems (CRSS), University of Texas at Dallas, Texas, U.S.A.
[3] Zurich Forensic Science Institute, Switzerland.
[4] Department of Computational Linguistics, University of Zurich, Switzerland.

1. Introduction

This paper presents an evaluation of the commercial forensic automatic speaker recognition system, VOCALISE (Voice Comparison and Analysis of the Likelihood of Speech Evidence) (Alexander et al. 2016), under conditions reflecting those of a real forensic case: *forensic_eval_01*. Full details of the evaluation rules, along with a description of the training data, testing data, and performance metrics, can be found in (Morrison and Enzinger 2016). VOCALISE is built with an 'open-box' architecture, offering the user choice between various feature extraction and speaker modelling approaches, and allowing the user to introduce their own development data at various points in the speaker modelling and comparison pipeline. This evaluation explores several ways in which the VOCALISE architecture can be applied to a realistic forensic comparison case.

2. Description and use of VOCALISE

VOCALISE is a forensic automatic speaker recognition system that allows a forensic practitioner to perform a speaker recognition comparison of two or more speech recordings to obtain a likelihood ratio. The processes available to the practitioner include the extraction of speaker-specific features from recordings of speech, the calculation of scores for pairwise comparisons of speech recordings, and the evaluation of likelihood ratios from the score distributions of same-speaker and different-speaker comparisons.

For this evaluation, VOCALISE is used to extract i-vector (Dehak et al. 2011) and x-vector (Snyder et al., 2018) speaker representations based on MFCC (Mel-frequency cepstral coefficient) features (Davis and Mermelstein 1980), and to compare these speaker representations using PLDA (probabilistic linear discriminant analysis) scoring (Prince and Elder 2007). Conversion of comparison scores to likelihood ratios using logistic regression calibration (Pigeon et al. 2000) is provided by Bio-Metrics, a software tool accompanying VOCALISE.

The VOCALISE 'front-end', which converts an audio recording into a set of MFCC features, and 'back-end', which compares vector representations of two recordings, is shared across the i-vector and x-vector approaches. The difference between the two approaches is in the

method of extracting a speaker representative vector, given a set of MFCC features. The i-vector approach relies on a factorisation of a GMM-MAP (Gaussian mixture modelling - maximum a posteriori), (Reynolds et al. 2000) speaker space, while the x-vector approach uses a DNN (deep neural network) 'embedding' to obtain a speaker representation.

Section 2.1 provides a general introduction to the relevant VOCALISE components. Specific information relating to feature and modelling parameters for this evaluation are provided in Sections 2.2—2.4.

2.1 Overview of VOCALISE components

2.1.1 System front-end: audio to features

The purpose of the system front-end is to extract information from the audio recording that is effective for speaker discrimination (Kinnunen and Li 2010). This process is referred to as feature extraction, and typically involves measuring the frequency characteristics of short segments across the file. MFCCs are a commonly employed feature, providing a representation of the short-term power spectrum of speech, weighted according to a perceptual scale of human hearing (Davis and Mermelstein 1980). MFCCs can be appended with their first and second derivatives, calculated over several frames (referred to as delta and delta-delta coefficients), as a means of incorporating temporal information (Furui 1986). To suppress the effects of non-speech variability, i.e., noise, normalisation is generally applied to the features. Cepstral Mean Subtraction (CMS) (Furui 1981) is way of removing convolutional noise, such as the effect of channel. Prior to speaker modelling, features corresponding to non-speech portions of the sample are discarded, and only those corresponding to speech are retained. This process, referred to as voice activity detection (VAD), typically discriminates between speech and non-speech based on the relative energy over short windows of the original speech sample (Kinnunen and Li 2010).

2.1.2 Speaker vector representation: i-vectors

The i-vector extraction process converts a set of MFCCs from an audio recording of arbitrary duration into a compact, fixed-length vector, in which most of the speaker variability is retained (Dehak et al. 2011), This process requires an i-vector extractor that has been trained with MFCCs from a large, independent set of training recordings.

To train the i-vector extractor, MFCCs from the training set are pooled to estimate the parameters of a GMM with a large number of components – a Universal Background Model (UBM). The UBM is then MAP-adapted (Reynolds et al. 2000) toward each of the training recordings given the relevant MFCCs. The resulting GMM-MAP speaker models are each converted into a supervector representation by stacking their mean components into a vector. The collected training supervectors are then factorised into largely speaker-dependent and -independent components; this is the key to obtaining a low-dimensional speaker representation, i.e. the i-vector. The speaker-independent component (represented by the UBM supervector), is first subtracted from the training supervectors, and the remaining speaker-dependent component is factorised into a low-rank Total Variability (TV) matrix and a set of total factors. These total factors, or i-vectors, can be viewed as low-dimensional representations of GMM-MAP supervectors that capture most of the important speaker variability. An i-vector can then be extracted for a new recording given a set of MFCCs, a UBM, and a trained TV matrix. Training the i-vector extractor is a data-hungry process;

typically, tens of thousands of recordings from thousands of speakers are used for training a UBM and TV matrix.

2.1.3 Speaker vector representation: x-vectors

The x-vector extraction process converts a set of MFCCs into a compact, fixed-length vector, in which most of the speaker variability is retained (Snyder et al., 2018). This process requires an x-vector extractor that has been trained with MFCCs from an independent set of training speakers.

The x-vector extractor is a feed-forward DNN architecture that consists of five frame-level layers that operate over varying time contexts, a statistics pooling layer, two recording-segment-level layers, and a softmax output layer (Snyder et al., 2018). The DNN is trained to classify a set of training speakers using short utterances (≈ 3 seconds) and speaker labels as input.

The first five layers model both static and temporal characteristics of the MFCCs; the first layer takes the MFCC frame at time t as input, along with a small temporal context (i.e., several additional MFCC frames centred on the current frame t). The second layer takes the output of the first layer as its input, and again includes a small temporal context, relative to the first layer. This process repeats through the five frame-level layers. The output of the fifth layer is passed to the statistics pooling layer, which produces an utterance-level representation by calculating the mean and standard deviation over all input MFCC frames from the current utterance. These statistics are passed through the two final, low-dimensional layers, and then the softmax output layer (which has the same dimensionality as the number of training speakers). Since the goal of the network is to provide representations for speakers not present in the training set, and to do so based on whole recordings, either of the two final utterance-level layers are suitable. Generally, the first of these two final layers (of dimension 512), is taken as the speaker embedding, i.e. the x-vector.

Training the x-vector extractor is a data-hungry process, requiring tens of thousands of recordings from thousands of speakers. It has been found that 'augmenting' the training set by including noised copies of the training recordings is beneficial (Snyder et al., 2018, McLaren et al., 2018). An established strategy is three-fold augmentation, which supplements each original recording in the training set with two noised copies.

2.1.4 System back-end: comparing vectors

Since x-vectors and i-vectors are fixed-length speaker representations, they could be directly compared using a simple distance metric, such as Cosine distance, for example. However, it is beneficial both to post-process the vectors to enhance their separability, and to utilise a model-based comparison metric, namely PLDA (probabilistic linear discriminant analysis) (Prince and Elder 2007).

Linear Discriminant Analysis (LDA) (McGlachan, 1992) is applied to the speaker vectors to enhance speaker separability while reducing the vector dimensionality. LDA uses a set of labelled training vectors to find a subspace in which the between-speaker variability is maximised and the within-speaker variability is minimised. The trained LDA transformation is used to project new vectors into this more discriminative subspace.

PLDA exploits knowledge of the most discriminative parts of an i-vector or x-vector to compare two vectors. With the Gaussian variant of PLDA, vectors are first length-normalised, and then factorised into largely speaker-dependent and –independent components (Garcia-Romero and Espy-Wilson, 2011). A set of labelled training vectors are used to learn the speaker-dependent vector subspace, which is subsequently used to compare new vectors.

2.1.5 VOCALISE systems

In VOCALISE, trained models and their associated parameters are stored as 'sessions' (Alexander et al. 2016). The session file for a system contains the front-end parameters, the UBM and TV matrix (in the case of an i-vector system), the trained DNN parameters (in the case of an x-vector system), and the back-end models for LDA and PLDA. Once a session is selected within the VOCALISE interface, the models and parameters associated with that session are used to compare test data.

Oxford Wave Research supplies pre-trained and optimised models with VOCALISE in the form of 'built-in' sessions that have been trained with tens of thousands of recordings from many speakers in different conditions. The user can also create custom sessions, either by adapting an existing session with supplementary data, or 'from-scratch', using exclusively their own data.

The following sections report the use of several different MFCC-based i-vector and x-vector VOCALISE configurations[1] on *forensic_eval_01*: built-in, condition-adapted, and from-scratch.

2.2 Built-in sessions

This section reports the use of built-in VOCALISE i-vector and x-vector sessions.

2.2.1 *i-vector session*

The built-in i-vector session '2017B-adaptable' was evaluated on *forensic_eval_01*. In this session, 15-dimensional MFCCs are extracted over 32 ms Hamming windows with 50% overlap. 24 Mel filterbanks in the range 1 Hz – 4,000 Hz are used for MFCC extraction. Features are appended with delta and delta-delta coefficients, CMS is applied globally, and silence frames are dropped according to instantaneous-SNR-based VAD.

The session training data consists of recordings of telephone and microphone speech from several thousand speakers, with an average recording duration of approximately three minutes. The full session training data set was used for training a UBM of 1024 components, a TV matrix of 400 dimensions, and LDA and PLDA models of 200 dimensions. No *forensic_eval_01* data was used in training the session.

Using this built-in session, a score was calculated for each test data comparison in the *forensic_eval_01* evaluation protocol. The test data consists of a total of 223 recordings from

[1] VOCALISE 2017B was used for all i-vector comparisons. VOCALISE 2019A-Beta-RC1 was used for all x-vector comparisons.

61 speakers, resulting in 111 same-speaker comparison scores and 9720 different-speaker comparison scores.

Score normalisation was applied to each score using a symmetric normalisation (S-norm) procedure (Shum et al. 2010). To apply S-norm, each test file was first compared with every *forensic_eval_01* training file (a total of 423 recordings from 105 speakers), resulting in a set of normalisation scores for each test file. The mean and standard deviation of each set of normalisation scores was calculated, resulting in a set of normalisation statistics for each test file. Then, given a score resulting from the comparison of two test files, two normalised scores were generated by independently applying the normalisation statistics for each test file to the score (normalisation was applied by subtracting the mean and dividing by the standard deviation). The symmetrically-normalised score was then given by the sum of the two normalised scores.

To transform the VOCALISE scores into log-likelihood ratios, logistic regression calibration (Pigeon et al. 2000) was applied with a leave-one-out cross-validation approach using Bio-Metrics 1.6 performance metrics software[2] (packaged with VOCALISE). In our approach, all of the scores originating from a particular speaker are transformed with calibration parameters estimated from all of the scores originating from the other speakers in the *forensic_eval_01* test set. This satisfies the evaluation requirement in (Morrison and Enzinger 2016) that, "cross validation must leave out all recordings from the speaker or speakers being tested (the one speaker for a same-speaker comparison or the two speakers for a different-speaker comparison), not just the two recordings actually being compared".

2.2.2 x-*vector session*

A built-in x-vector session was evaluated on *forensic_eval_01*. In this session, 20-dimensional MFCCs are extracted over 32 ms Hamming windows with 50% overlap. 24 Mel filterbanks in the range 1 Hz – 4,000 Hz are used for MFCC extraction. CMS is applied globally, and silence frames are dropped according to instantaneous-SNR-based VAD.

The session training data consists of telephone and microphone speech from several thousand speakers. Three-fold augmentation (Snyder at al., 2018) was applied to the training set: for each recording, two augmentations were randomly selected from a choice of babble, music, noise, and reverberation, resulting in three 'versions' of each original training recording. Random selections of babble, music, and noise were added to the original audio signals at an SNR of 5 dB, based on recommendations in McLaren et al., 2018. To add reverb, the original training recordings were convolved with random selections of simulated room impulses (Snyder et al., 2015). No *forensic_eval_01* data was used in training this session.

The DNN architecture is the same as Synder et al., 2018. The first utterance-level layer (following the statistics pooling layer) of 512 dimensions is taken as the speaker embedding. LDA and PLDA models of 150 dimensions are trained using x-vectors obtained from the trained network embedding layer. The full session training data set was used for training the DNN, LDA and PLDA models.

[2] Oxford Wave Research Ltd., http://www.oxfordwaveresearch.com/products/bio-metrics

Using this built-in session, a score was calculated for each test data comparison in the *forensic_eval_01* evaluation protocol, and symmetric score normalisation and leave-one-out cross-validation calibration were applied, all in the same manner as Section 2.2.1

2.3 Condition-adapted sessions

As the recording conditions encountered in forensic cases vary widely, it is not conceivable that a commercial forensic automatic speaker recognition system will have had sight of all possible conditions. Therefore, the system may not perform optimally in the specific conditions of the case. VOCALISE provides the capability to adapt the underlying built-in models to the conditions of the case, while also taking into account that the quantity of case-specific data available is usually relatively small compared to the quantity of data used to train the i-vector or x-vector system from scratch.

This section reports the use of the VOCALISE built-in i-vector ('2017B-adaptable') and x-vector sessions, both adapted using the *forensic_eval_01* training set.

Condition-adaptation in VOCALISE allows an existing (e.g., built-in) session to be adapted to a new condition using new, supplementary data. The recordings provided for condition-adaptation are used to update the LDA and PLDA models in the existing session. Thus, the adaptation can be applied to both i-vector and x-vector systems as follows: first, vectors are extracted for the new data. A weighted interpolation of the new vector and existing vector statistics is then used to update the LDA transformation matrix. All (new and existing) vectors are transformed with the updated LDA model and are then used to re-estimate the PLDA model.

For this evaluation, an augmented *forensic_eval_01* training set was used for condition-adaptation. Augmented training recordings were generated by applying each of the four augmentations detailed in Section 2.2.2 (babble, music, noise, and reverberation) to each of the original training recordings. The combined set of 423 original training recordings and 1692 augmented training recordings was then used for condition-adaptation.

After condition-adaptation, the adapted i-vector and x-vector sessions differ from the built-in sessions (described in Section 2.2) only in terms of their LDA and PLDA models. Using the adapted sessions, a score was calculated for each test data comparison in the *forensic_eval_01* evaluation protocol. As the *forensic_eval_01* training set was used for condition-adaptation, score normalisation was not applied. In our experience, condition-adaptation continues to improve performance as the size of the training set increases – we therefore decided against dividing up the training data to apply both condition-adaptation and score normalisation. Cross-validation calibration was applied as in Section 2.2.1.

2.4 From-scratch *forensic_eval_01* sessions

In VOCALISE, full i-vector and x-vector pipelines can be trained from-scratch with user-provided data. This section reports the use of i-vector and x-vector VOCALISE sessions

trained using the *forensic_eval_01* training set exclusively[3]. The *forensic_eval_01* training set contains 423 recordings from 105 speakers, whereas the built-in training sets (Section 2.2) contain thousands of speakers. This reduction in the number of training speakers necessitated the use of smaller modelling and back-end parameters for the from-scratch systems than for the built-in systems.

In da Silva and Medina 2017, a from-scratch i-vector system trained with the *forensic_eval_01* training set was evaluated and shown to perform well. Therefore, a decision was made to select from-scratch i-vector session parameters that aligned closely with that system. They are as follows: 14-dimensional MFCCs are extracted over 32 ms Hamming windows with 50% overlap. 24 Mel filterbanks in the range 300–3,300 Hz are used for MFCC extraction. Features are appended with delta coefficients. CMS is applied globally, and silence frames are dropped according to instantaneous-SNR-based VAD. All *forensic_eval_01* training recordings were used for training a UBM (1024 components), a TV matrix (200 dimensions), and LDA and PLDA models (both 50-dimensional).

An x-vector session was trained using an augmented *forensic_eval_01* training set. Augmented training recordings were generated by applying each of the four augmentations detailed in Section 2.2.2 (babble, music, noise, and reverberation) to each of the original training recordings. The combined set of 423 original training recordings and 1692 augmented training recordings was used for training the session. The front-end features are consistent with the built-in x-vector session in Section 2.2.2: 20-dimensional MFCCs are extracted over 32 ms Hamming windows with 50% overlap. 24 Mel filterbanks in the range 1 Hz – 4,000 Hz are used for MFCC extraction. CMS is applied globally, and silence frames are dropped according to instantaneous-SNR-based VAD. The DNN architecture is again the same as Synder et al., 2018. Given the smaller quantity of data, the size of the embedding layer is reduced to 256, followed by 50-dimensional LDA and PLDA models.

Using the from-scratch *forensic_eval_01* sessions, a score was calculated for each test data comparison in the *forensic_eval_01* evaluation protocol. As the *forensic_eval_01* training set was used for session training, score normalisation was not applied. Cross-validation calibration was applied as in Section 2.2.1.

3 Results

The evaluation results for each of the six systems (built-in, condition-adapted, and from-scratch variants of both i-vector and x-vector systems) are summarised in Table 1. For definitions of C_{llr}^{pooled}, C_{llr}^{mean} and C_{llr}^{cal}, refer to Morrison and Enzinger 2016.

	System variant	C_{llr}^{pooled}	C_{llr}^{mean}	95% CI	C_{llr}^{min}	C_{llr}^{cal}	EER
i-vector	built-in	0.462	0.447	0.530	0.416	0.046	0.115
	adapted	0.267	0.230	1.178	0.239	0.029	0.070
	from-scratch	0.462	0.426	0.965	0.393	0.069	0.110
x-vector	built-in	0.303	0.275	0.775	0.247	0.056	0.079

[3] In our experience, the quantity of data in the *forensic_eval_01* training set is not sufficient to train an effective x-vector system; augmentation was therefore applied to the training set, increasing the quantity and diversity of the data.

adapted	0.246	0.213	1.040	0.189	0.057	0.053
from-scratch	0.447	0.379	0.836	0.419	0.028	0.135

Table 1. Exact values of the accuracy and precision metrics.

Figure 1 provides a graphical representation of the C_{llr} metrics in Table 1. For each of the six systems, Figures 2—5 provide Tippett plots, Detection Error Trade-Off (DET) curves, and Empirical Cross Entropy (ECE) plots respectively.

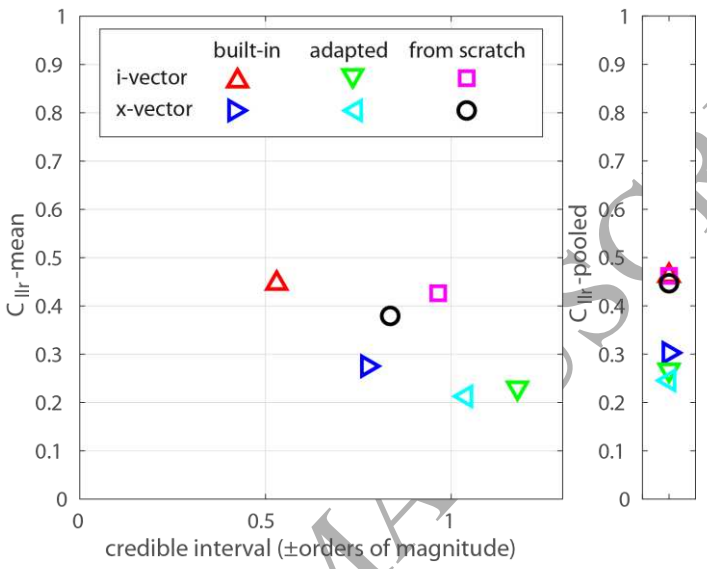

Figure 1. Plot showing C_{llr}^{mean} versus 95% CI (left panel) and C_{llr}^{pooled} (right panel).

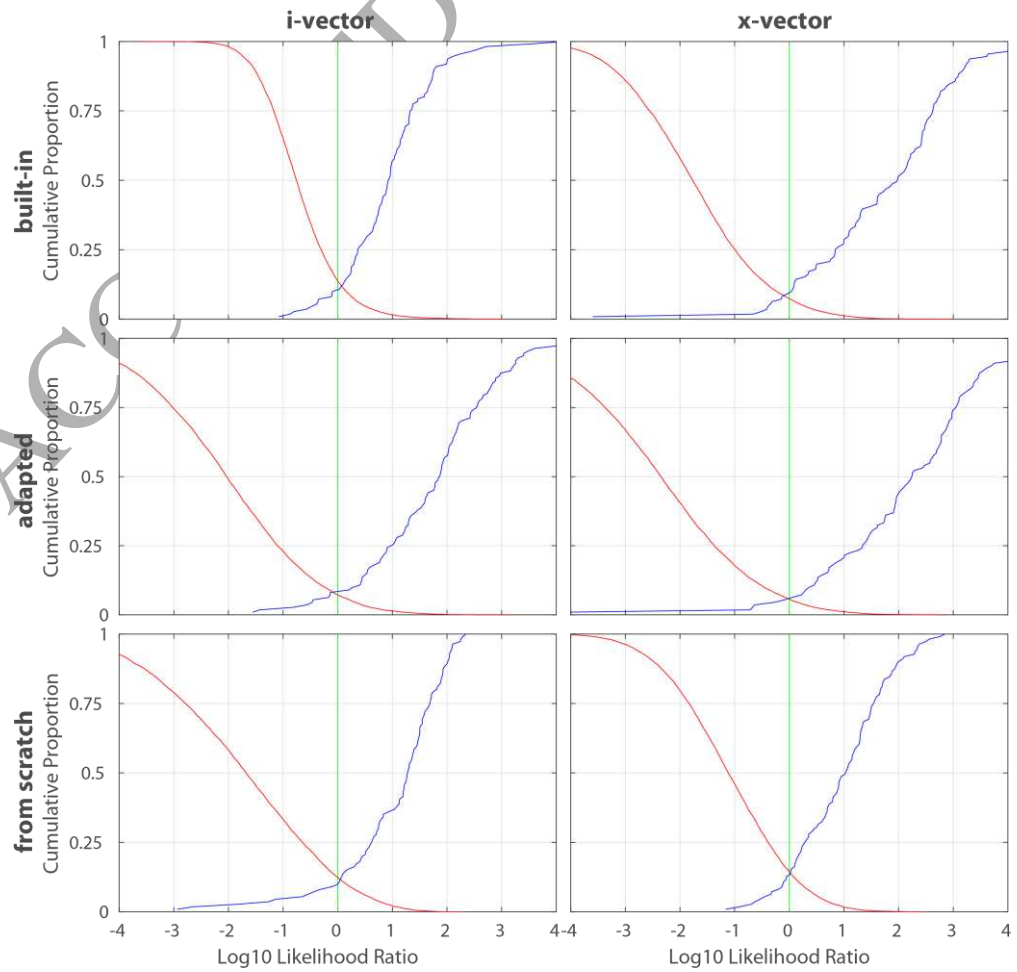

Figure 2. Tippett plots (no precision) of the results of the tested systems. Left panels show tests using i-vectors, right panels those using x-vectors. The first row shows built-in sessions, the second shows condition-adapted sessions, and the third shows from-scratch sessions.

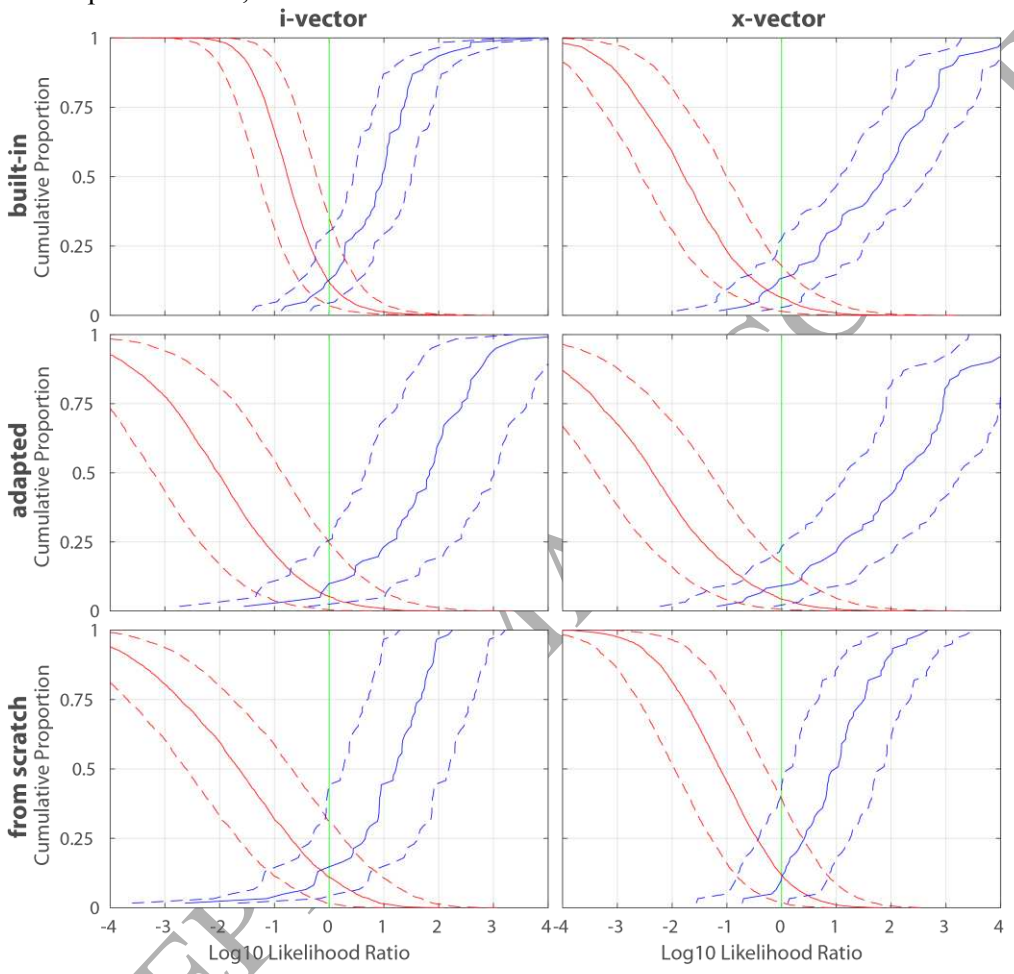

Figure 3. Tippett plots (with precision) of the results of the tested systems. Left panels show tests using i-vectors, right panels those using x-vectors. The first row shows built-in sessions, the second shows condition-adapted sessions, and the third shows from-scratch sessions.

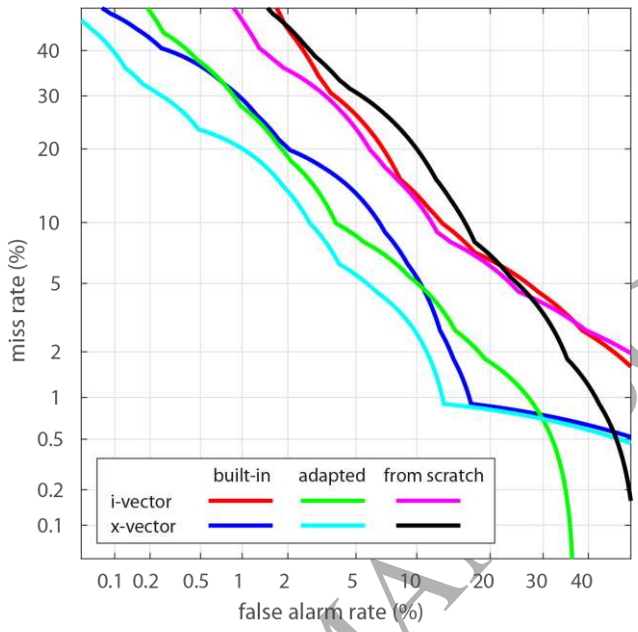

Figure 4. Detection Error Trade-off (DET) plot.

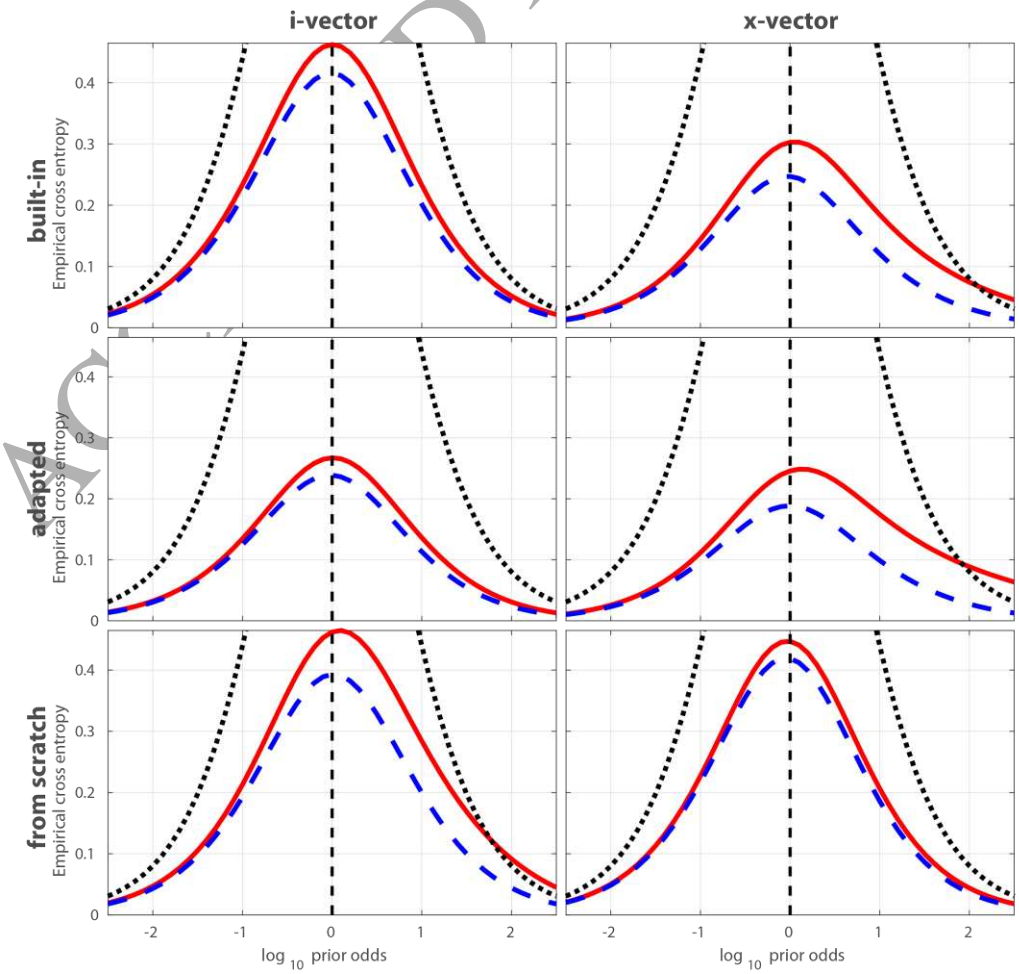

Figure 5. Empirical Cross Entropy (ECE) plots of the results of the tested systems. Left panels show tests using i-vectors, right panels those using x-vectors. The first row shows built-in sessions, the second shows condition-adapted sessions, and the third shows from-scratch sessions.

4 Discussion and conclusion

This study evaluated VOCALISE i-vector and x-vector speaker recognition systems under conditions reflective of a real forensic case. Three variants of each system were considered, evaluating different ways in which case-relevant user-provided data can be used within VOCALISE.

Comparing i-vector and x-vector systems in terms of pure discrimination (measured by C_{llr}^{min} and EER) and accuracy (discrimination and calibration, measured by C_{llr}^{pooled} and C_{llr}^{mean}) (van Leeuwen and Brümmer, 2007), it can be seen that the x-vector system outperforms the i-vector system for both the built-in and condition-adapted variants. While the from-scratch i-vector system outperforms the from-scratch x-vector system, we expect that this is due to training data insufficiency. The best overall performance is achieved by the condition-adapted x-vector system.

The built-in and condition-adapted variants of the i-vector and x-vector systems demonstrate how case-relevant data can be used with a pre-trained system (i.e. a VOCALISE session); in the built-in case, case-relevant data is used for score-normalisation, and in the condition-adapted case, for LDA/PLDA adaptation. For both i-vector and x-vector systems, the condition-adapted variant achieves better performance, indicating that this is the more effective use of the case-relevant data. The from-scratch variant of the i-vector system achieves similar performance to the i-vector built-in variant, and outperforms the from-scratch x-vector variant. We note that a from-scratch system will not typically be a feasible option for a forensic practitioner in a real case, as the quantity of case-relevant data available is unlikely to match (or exceed) the size of the *forensic_eval_01* training set. In this scenario, score-normalisation and condition-adaptation provide an effective way to adapt a pre-trained system to the conditions of the case.

Conflict of interest

The authors declare no conflict of interest.

References

Alexander, A., Forth, O., Atreya, A. A., Kelly, F., 2016. VOCALISE: A Forensic Automatic Speaker Recognition System Supporting Spectral, Phonetic, and User-Provided Features. In Odyssey 2016.

Biometrics 1.6, 2017. Performance Metrics software, Oxford Wave Research Ltd., http://www.oxfordwaveresearch.com/products/bio-metrics

Davis, S., Mermelstein, P., 1980. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4): 357–66.

Dehak, N., Kenny, P.J., Dehak, E., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Transactions on Acoustics, Speech, and Signal Processing. 19(14): 788–798.

Furui, S., 1981. Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech, and Signal Processing, 29: 254–272.

Furui, S., 1986. Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Transactions on Acoustics, Speech, and Signal Processing, 34: 52–59.

Garcia-Romero, D., Espy-Wilson, C.Y., 2011, Analysis of i-vector Length Normalization in Speaker Recognition Systems. In Interspeech 2011: 249--252

Kelly, F., Alexander, A., Forth, O., Kent, S., Lindh, J., Åkesson, J., 2016. Identifying Perceptually Similar Voices with a Speaker Recognition System Using Auto-Phonetic Features. In Interspeech 2016: 1567–68.

Kinnunen, T., Li, H., 2010. An overview of text-independent speaker recognition: From features to supervectors, Speech Communication, 52(1): 12—40.Morrison, G. S., Enzinger, E., 2016. Multi-Laboratory Evaluation of Forensic Voice Comparison Systems under Conditions Reflecting Those of a Real Forensic Case. Speech Communication, 85: 119–26.

van Leeuwen, D.A., Brümmer, N., 2007. An Introduction to Application-Independent Evaluation of Speaker Recognition Systems. Speaker Classification I, 330—353, Speaker Classification I, Lecture Notes in Computer Science, vol 4343. Springer, Berlin, Heidelberg, ISBN: 978-3-540-74186-2

Morrison, G. S., Enzinger, E., 2016. Multi-Laboratory Evaluation of Forensic Voice Comparison Systems under Conditions Reflecting Those of a Real Forensic Case. Speech Communication, 85: 119–26.

McLachlan, G. J., 1992, Discriminant analysis and statistical pattern recognition. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York, ISBN: 0-471-61531-5.

McLaren, M., Castán, D., Nandwana, M.K., Ferrer, L., Yılmaz, E., 2018, How to Train Your Speaker Embeddings Extractor, In Odyssey 2018: 327—334.

Pigeon, S., Druyts, P., Verlinde, P., 2000. Applying Logistic Regression to the Fusion of the NIST ‘99 1-Speaker Submissions. Digital Signal Processing, 10 (1–3): 237–248.

Prince, S., Elder, J., 2007. Probabilistic Linear Discriminant Analysis for Inferences about identity. In IEEE 11th International Conference on Computer Vision (ICCV), 2007: 1–8.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker Verification using Adapted Gaussian Mixture Models. Digital Signal Processing. 10 (1–3): 19–41.

Shum, S., Dehak, N., Dehak, R., Glass, J. R., 2010, Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification. In Odyssey 2010.

da Silva, D. G., Medina, C. A., 2017. Evaluation of MSR Identity Toolbox under conditions

reflecting those of a real forensic case (*forensic_eval_01*). Speech Communication, 94: 42–49

Snyder, D., Chen, G., Povey, D., 2015, MUSAN: A Music, Speech, and Noise Corpus, arXiv: 1510.08484v1

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018, X-Vectors: Robust DNN Embeddings for Speaker Recognition, In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018: 5329-5333.